FINAL SCRIPT: APPROVED FOR FILMING

Submission ID #: 68854
Scriptwriter Name: Pallavi Sharma
Project Page Link: https://review.jove.com/account/file-uploader?src=21000518

Title: Rapid In-Silico Battery Electrolyte Electrochemical Reaction Generation using 3T-VASP Multi-Scale Energy Minimization

Authors and Affiliations:
Jonathan P. Mailoa[1,2*], Xin Li[3], Zhengmi Tang[2], Jineng Ren[1,2], Mingyang Ni[1,2], Shengyu Zhang[4*]

[1]College of Computer Science and Artificial Intelligence, Wenzhou University
[2]Wenzhou University Artificial Intelligence and Advanced Manufacturing Institute
[3]Tencent Quantum Laboratory, Tencent, Shenzhen
[4]Tencent Quantum Laboratory, Tencent, Hong Kong SAR

Corresponding Authors:
Jonathan P. Mailoa (jpmailoa@alum.mit.edu)
Shengyu Zhang (shengyzhang@tencent.com)

Email Addresses for All Authors:
Xin Li (maskerli@tencent.com)
Zhengmi Tang (tzm@aiami.tech)
Jineng Ren (renjineng@aiami.tech)
Mingyang Ni (myni@aiami.tech)
Jonathan P. Mailoa (jpmailoa@alum.mit.edu)
Shengyu Zhang (shengyzhang@tencent.com)

Author Questionnaire

1. **Microscopy**: Does your protocol require the use of a dissecting or stereomicroscope for performing a complex dissection, microinjection technique, or something similar? **NO.**

2. **Software:** Does the part of your protocol being filmed include step-by-step descriptions of software usage? **YES**

3. **Filming location:** Will the filming need to take place in multiple locations? **No**

Current Protocol Length
Number of Steps: 14
Number of Shots: 28

Introduction

Videographer: Obtain headshots for all authors available at the filming location.

1.1. **Xin Li:** This research uses 3T-VASP to study lithium-ion battery electrolyte reaction pathways, analyzing byproducts and SEI to tackle experimental challenges and set up workflows for new systems.

1.1.1. INTERVIEW: Named talent says the statement above in an interview-style shot, looking slightly off-camera. *Suggested B roll: Figure 1*

What are the current experimental challenges?

1.2. **Xin Li:** Existing experimental and computational methods struggle to study precise electrolyte reaction pathways due to the many types and hard-to-analyze complex SEI.

1.2.1. INTERVIEW: Named talent says the statement above in an interview-style shot, looking slightly off-camera.

What research gap are you addressing with your protocol?

1.3. **Xin Li:** Ab-initio methods like DFT are slow for finding rare electrochemical byproducts, needing many steps. This paper fills the gap with a fast, multi-scale approach that uses far fewer DFT steps.

1.3.1. INTERVIEW: Named talent says the statement above in an interview-style shot, looking slightly off-camera.

Videographer: Obtain headshots for all authors available at the filming location.

Testimonial Questions (OPTIONAL):

Videographer: Please capture all testimonial shots in a wide-angle format with sufficient headspace, as the final videos will be rendered in a 1:1 aspect ratio. Testimonial statements will be presented live by the authors, sharing their spontaneous perspectives.

How do you think publishing with JoVE will enhance the visibility and impact of your research?

1.4. **Xin Li,** I've never submitted to video journals, which is novel compared to past experiences, and videos help users use our method more easily. (authors will present their testimonial statements live)

1.4.1. INTERVIEW: Named talent says the statement above in an interview-style shot, looking slightly off-camera.

Protocol

2. Setting up a Computing Environment for Running 3T-VASP

Demonstrator: Xin Li

2.1. To begin, create a new conda environment named 3T with Python version 3.11 by entering the appropriate command in the terminal **[1]**. Activate the 3T environment using the activate command **[2]**. Then, use conda to install Git from the conda-forge channel **[3]**.

2.1.1. WIDE: Talent entering the command conda create --name 3T python=3.11 in the terminal window.

2.1.2. SCREEN: Updated_JoVE_68854_2.1.mp4: 00:29-00:35

2.1.3. SCREEN:Updated_JoVE_68854_2.1.mp4 00:49-00:56, 01:20-01:23, 01:28-01:30

2.2. Clone the 3T-VASP *(Three-T-Vasp)* GitHub *(Git-Hub)* repository using the URL **[1-TXT]**. Then, enter the cloned repository directory named External_3T *(External-Underscore-Three-T)* **[2]**.

2.2.1. SCREEN: Updated_JoVE_68854_2.2.mp4**:** 00:10-00:26, 01:50-01:53**TXT: https://github.com/jpmailoa/External_3T.git into the terminal**

2.2.2. SCREEN:Updated_JoVE_68854_2.2.mp4**:** 02:16-02:22.

2.3. Use conda to install Mamba from the conda-forge channel **[1]**. Then, use Mamba to install the required libraries for the 3T conda environment **[2]**.

2.3.1. SCREEN: Updated_JoVE_68854_2.3.mp4: 00:00-00:12

2.3.2. SCREEN: Updated_JoVE_68854_2.3.mp4**:** 01:53-02:00.

2.4. Next, install the software GROMACS *(Gro-Macs)* and InterMol *(Inter-Mol)* within the 3T environment **[1]**.

2.4.1. SCREEN: Updated_JoVE_68854_2.4.mp4: 00:09-00:14

2.4.2. SCREEN: Updated_JoVE_68854_2.4.mp4: 00:23-00:25, 00:57-01:01

2.5. After installing the modified version of InterMol *(Inter-Mol)* in the 3T environment, open the Python script file named calculator_3T_VASP.py *(calculator- -Three-T- -VASP-Dot-P-Y)* located in the utils directory using a code editor **[1]**. Scroll to the run VASP function and locate the default OS *(O-S)* system call used to execute the VASP software

[2]. Then, modify the command line to reflect the path of the user's VASP executable and specify the computing resources as needed **[3]**.

2.5.1. SCREEN: Updated_JoVE_68854_2.5.mp4: 00:00-00:18

2.5.2. SCREEN: Updated_JoVE_68854_2.3.mp4: 00:18-00:38

2.5.3. SCREEN: Updated_JoVE_68854_2.3.mp4**:** 00:43-00:59

2.6. In the Linux terminal, verify that non-Python third-party utilities required by 3T are available by executing the commands for gmx *(G-M-X)*, wget *(W-Get)*, unzip, and packmol *(Pack-Mol)* **[1]**.

2.6.1. SCREEN: Updated_JoVE_68854_2.6.mp4: 00:06-00:10, 00:14-00:17, 00:24-00:26, 00:37-00:40

2.7. To test if 3T-VASP has been configured correctly, use Python to run randomize_3T_bulk_electrolyte_reduction.py *(randomize- -3-T- -bulk- -electrolyte- -reduction-Dot-P-Y)* from the terminal **[1]**. Open another terminal to monitor the output file named default.log as the simulation runs **[2]**.

2.7.1. SCREEN: Updated_JoVE_68854_2.7.mp4: 00:00-00:10

2.7.2. SCREEN: Updated_JoVE_68854_2.7.mp4: 00:12-00:19, 00:52-00:57

3. Preparing 3T-VASP input files and Running 3T-FF and 3T-VASP Energy Minimizations

3.1. Name the 3T-VASP lattice file using the <file_name>.vasp *(File-Name-Vasp)* convention and save each file into an appropriate subfolder inside the input directory **[1]**.

3.1.1. SCREEN: Updated_JoVE_68854_3.1&3.2.mp4 00:07-00:12, 00:26-00:36

3.2. Create override files for each periodic box lattice structure with the file extension override *(Dot-Override)* **[1]**. Place them in the same subfolder under the input directory as the corresponding .vasp files **[2]**. Then, save these files as JSON *(J-Son)* dictionaries with the required keys movable_group *(movable-underscore-group)* and atom_charge_proximity *(Atom-Underscore-Charge-Underscore-Proximity)* **[2]**.

3.2.1. SCREEN: Updated_JoVE_68854_3.1&3.2.mp4 :01:17-01:23.

3.2.2. SCREEN: Updated_JoVE_68854_3.1&3.2.mp4: 01:45-02:00

3.2.3. SCREEN: Updated_JoVE_68854_3.1&3.2.mp4: 01:23-01:35

3.3. To generate molecule structure files in the XYZ *(X-Y-Z)* format, name them using the <file_name>.xyz *(File-Name-Dot-X-Y-Z)* pattern **[1]** and place these files directly inside the input folder **[2]**.

3.3.1. SCREEN: Updated_JoVE_68854_3.3.mp4: 00:05-00:24

3.3.2. SCREEN: Updated_JoVE_68854_3.1&3.2.mp4: 00:35-00:40

3.4. Prepare the template VASP input files required for each 3T-VASP step, including INCAR *(In-Car)*, KPOINTS *(K-points)*, and POTCAR *(Pot-Car)* files **[1]**. In the INCAR file, ensure the molecular dynamics step count parameter NSW is either omitted or explicitly set to zero **[2]**.

3.4.1. SCREEN: New_JoVE_68854_3.4.mp4: 00:58-01:02, 01:30-01:35, 02:20-02:23

3.4.2. SCREEN: New_JoVE_68854_3.4.mp4: 04:04-04:11, 04:26-04:28

3.5. Next, prepare a 3T configuration file that references and integrates all the created input files **[1]**.

3.5.1. SCREEN: New_JoVE_68854_3.5.mp4: 00:15-00:21, 00:59-01:12

3.6. To perform a single 3T-FF *(3-T-F-F)* or 3T-VASP trajectory generation, launch Python in the terminal **[1]**. Import the main function from the main_run_utils *(Main-Run-Utlis)* module **[2]**. Execute the trajectory generation using a specific configuration file **[3]**. Monitor the progress by viewing the default.log file in a separate terminal **[4]**.

3.6.1. SCREEN: Updated_New_JoVE_68854_3.6.mp4: 00:00-00:07

3.6.2. SCREEN: Updated_New_JoVE_68854_3.6.mp4: 00:07-00:17

3.6.3. SCREEN:Updated_New_JoVE_68854_3.6.mp4: 00:24-00:28, 01:08-01:10

3.6.4. SCREEN: Updated_New_JoVE_68854_3.6.mp4: 01:10--01:30

3.7. To perform large-scale trajectory generation, write a short function to replace specific phrases in the config template file, generating new config files to produce different 3T-VASP trajectories **[1]**.

3.7.1. SCREEN: New_JoVE_68854_3.7.mp4: 00:09-00:30

Results

4. Results

4.1. A gradual dispersion of electrolyte molecules was observed inside the periodic boundary condition box during the 3T-FF phase, without any chemical reactions occurring **[1]**, confirming a stable force field setup **[2]**, as seen across the three snapshots **[3]**.

4.1.1. LAB MEDIA: Figure 9A. *Video editor: Highlight the first panel where molecules are still clustered together*.

4.1.2. LAB MEDIA: Figure 9A. V*ideo editor: Highlight the middle panel where the molecular arrangement shows partial dispersion.*

4.1.3. LAB MEDIA: Figure 9A. *Video editor: Highlight the third panel where the molecules appear evenly distributed throughout the space*.

4.2. During the 3T-VASP phase, electrochemical reduction of ethylene carbonate into ethylene and carbonate anion was commonly observed **[1]**, while a secondary reaction producing ethane-1,2-diolate anion and carbon monoxide occurred less frequently **[2]**.

4.2.1. LAB MEDIA: Figure 9B. *Video editor: Highlight the first two boxes present on the left side*

4.2.2. LAB MEDIA: Figure 9B. *Video editor: Highlight the last two boxes*

1. Conda

- **Pronunciation link**: YouTube tutorial on pronouncing "Conda" (quickpronounce.site, youtube.com)
- **IPA**: /ˈkɒn.də/
- **Phonetic Spelling**: *KON-duh*

---

2. Mamba

- **Pronunciation link**: (Standard English—no specialized source found)
- **IPA**: /ˈmæm.bə/
- **Phonetic Spelling**: *MAM-buh*

---

3. GROMACS (Gr-oh-macks)

- **Pronunciation link**: Common software name; standard pronunciation
- **IPA**: /grəˈmɑːks/
- **Phonetic Spelling**: *gruh-MAHKS*

---

4. InterMol (In-ter-mol)

- **Pronunciation link**: Pronunciation follows standard English compound rules
- **IPA**: /ˈɪn.tər.mɒl/
- **Phonetic Spelling**: *IN-ter-mol*

---

5. VASP (software acronym: V-A-S-P)

- **Pronunciation**: Typically spelled out rather than spoken as a word
- **IPA (spelled-out)**: /ˈviː.eɪ.esˈpiː/
- **Phonetic Spelling**: *VEE-AY-ESS-PEE*

---

6. Packmol (Pack-mol)

- **Pronunciation link**: Common package name in computational chemistry
- **IPA**: /pækˈmɒl/
- **Phonetic Spelling**: *pak-MOL*

---

7. XYZ (format name spelled out)

- **Pronunciation**: Letters read individually
- **IPA**: /ɛks.waɪˈziː/
- **Phonetic Spelling**: *eks-WYE-zee*

---

8. JSON (Jay-son)

- **Pronunciation link**: Common programming term
- **IPA**: /ˈdʒeɪ.sən/
- **Phonetic Spelling**: *JAY-suhn*
